IN THE CLAIMS:

1. (Currently Amended) An SMBus message handler comprising:

a memory ~~(202)~~ configured to store microcode comprising at least two programs ~~(210, 211, 212)~~ each for handling a bus command protocol and comprising at least one instruction;

an interface to a register ~~(SMB_PRTCL)~~ configured to identify a starting address of a program in said memory;

an instruction fetch unit ~~(203)~~ configured to read an instruction at an address in said memory ~~(202);~~ said address being specified by a program counter ~~(pc)~~; and

a finite-state machine ~~(201)~~ configured to receive and ~~interpreting~~ interpret the instructions read by said instruction fetch unit ~~(203)~~ and ~~for managing~~ manage the data transfer between an SMBus ~~(213, 214)~~ interface[[,]] and a register set ~~(208)~~ in compliance with said instructions read from said memory.

2. (Original) The SMBus message handler of claim 1, wherein said register set complies with the ACPI specification.

3. (Currently Amended) The SMBus message handler of claim 1, further comprising an address register array ~~(207)~~ comprising a plurality of starting addresses of programs stored in said memory, said register ~~(SMB_PRTCL)~~ comprising an offset for pointing at a specific register in said address register array.

4. (Currently Amended) The SMBus message handler of claim 2, further comprising a buffer pointer register ~~(206)~~ for pointing at one of a plurality of data registers ~~(SMB_DATA)~~; said finite-state machine ~~(201)~~ transferring data read from the SMBus ~~(213, 214)~~ interface to the data register at which said buffer pointer register ~~(206)~~ points if said finite-state machine ~~(201)~~ interprets a ~~"receive data to"~~ reception instruction; said finite-state machine ~~(201)~~ transferring the data read

from the data register at which said buffer pointer register points to said SMBus ~~(213, 214)~~ interface if said finite-state machine ~~(201)~~ interprets a ~~"transmits data from"~~ transmission instruction.

5. (Currently Amended) The SMBus message handler of claim 4, wherein the finite-state machine causes said buffer pointer register ~~(206)~~ to be incremented each time a ~~"transmit data to"~~ reception instruction or a ~~"transmit data from"~~ transmission instruction is executed.

6. (Currently Amended) The SMBus message handler of claim 1, further comprising a loop counter ~~(204)~~ for storing the value of a block counter register ~~SMB_BCNT~~ in said loop counter ~~(204)~~ if the finite-state machine executes an instruction to ~~a~~ "transmit data from said block counter register ~~SMB_BCNT" instruction;~~ to said SMBus interface, said loop counter ~~(204)~~ being decremented each time a data byte is transmitted to said SMBus ~~(213, 214)~~ interface while an instruction to ~~a~~ "transmit data from said block counter register" ~~instruction~~ to said SMBus interface is executed, wherein ~~and~~ the ~~"transmit data from"~~ instruction to transmit data from said block counter register to said SMBus interface is ~~be~~ completed when the value of said loop counter ~~(204)~~ reaches zero.

7. (Currently Amended) The SMBus message handler of claim 1, further comprising a loop counter ~~(204)~~ and a block count register ~~(SMB_BCNT)~~ both for storing a byte received from said SMBus ~~(213, 214)~~ interface if the finite-state machine ~~(201)~~ executes an instruction to ~~a "~~ receive data ~~to~~ at said block counter register ~~SMB_BCNT" instruction~~ from said SMBus interface, said loop counter ~~(204)~~ being decremented each time a data byte is transmitted to or received from said SMBus ~~(213, 214)~~ interface while an instruction to ~~a "received~~ receive data ~~to"~~ ~~instruction~~ at said block counter register from said SMBus interface is executed, wherein ~~and~~ the ~~"received data to"~~ instruction to receive data at said block counter register from said SMBus interface is ~~being~~ completed when the value of said loop counter ~~(204)~~ reaches zero.

8. (Currently Amended) The SMBus message handler of claim 1, wherein each instruction comprises one bit ~~(304)~~ indicating as to whether or not an instruction is the last instruction in the program.

9. (Currently Amended) The SMBus message handler of claim 1, wherein each instruction comprises one bit ~~(305)~~ indicating as to whether an instruction is to be executed only once or this instruction is to be executed repeatedly until a loop counter ~~(204)~~ becomes zero, wherein said loop counter is decremented each time an instruction is executed repeatedly.

10. (Currently Amended) An integrated circuit chip for transmitting and receiving data over ~~a~~ an SMBus, the integrated circuit chip comprising:

an interface to a memory ~~(202)~~ configured to store microcode comprising at least two programs ~~(210, 211, 212)~~ each for handling a bus command protocol and comprising at least one instruction;

an interface to a register ~~(SMB_PRTCL)~~ configured to identify a starting address of a program in said memory;

an instruction fetch unit ~~(203)~~ configured to read an instruction at an address in said memory ~~(202)~~; said address being specified by a program counter (pc); and

a finite-state machine ~~(201)~~ configured to receive and interpret the instructions read by said instruction fetch unit ~~(203)~~ and ~~for managing~~ to manage the data transfer between an SMBus ~~(213, 214)~~ interface, and a register set ~~(208)~~ in compliance with said instructions read from said memory.

11. (Original) The integrated circuit chip of claim 10, wherein said register set complies with the ACPI specification.

12. (Currently Amended) The integrated circuit chip of claim 10, further comprising an address register array ~~(207)~~ comprising a plurality of starting addresses of programs stored in said memory, said register ~~(SMB_PRTCL)~~ comprising an offset for pointing at a specific register in said address register array.

13. (Currently Amended) The integrated circuit chip of claim 11, further comprising a buffer pointer register ~~(206)~~ for pointing at one of a plurality of data registers ~~(SMB_DATA)~~ comprised in register set ~~(208)~~; said finite-state machine ~~(201)~~ transferring data read from the SMBus ~~(213, 214)~~ interface to the data register at which said buffer pointer register ~~(206)~~ points if said finite-state machine (201) interprets a ~~"receive~~ data reception ~~to"~~ instruction[[;]], said finite-state machine (201) transferring the data read from the data register at which said buffer pointer register points to said SMBus ~~(213, 214)~~ interface if said finite-state machine ~~(201)~~ interprets a ~~"transmits~~ data transmission ~~from"~~ instruction.

14. (Currently Amended) The integrated circuit chip of claim 13, wherein the finite-state machine ~~(201)~~ causes said buffer pointer register ~~(206)~~ to be incremented each time a ~~"transmit~~ data reception instruction ~~to"~~ or a ~~"transmit~~ data transmission instruction ~~from"~~ instruction is executed.

15. (Currently Amended) The integrated circuit chip of claim 10, further comprising a loop counter ~~(204)~~ for storing the value of a block counter register SMB_BCNT in said loop counter ~~(204)~~ if the finite-state machine executes an instruction to ~~a~~ "transmit data from said block counter register ~~SMB_BCNT" instruction~~ to said SMBus interface[[;]], said loop counter ~~(204)~~ being decremented each time a data byte is transmitted to said SMBus ~~(213, 214)~~ interface while an instruction to ~~a~~ "transmit data from said block counter register ~~" instruction~~ to said SMBus interface is executed, wherein ~~and~~ the ~~"transmit data from"~~ instruction to transmit data from said block counter register to said SMBus interface is ~~be~~ completed when the value said loop counter ~~(204)~~ reaches zero.

16. (Currently Amended) The integrated circuit chip of claim 10, further comprising a loop counter ~~(204)~~ and a block count register ~~(SMB_BCNT)~~ comprised in said register set ~~(208)~~ both for storing a byte received from said SMBus ~~(213, 214)~~ interface if the finite-state machine ~~(201)~~ executes an instruction to ~~a~~ "receive data ~~to~~ at said block counter register ~~SMB_BCNT" instruction~~ from said SMBus interface, said loop counter ~~(204)~~ being decremented each time a data byte is

transmitted to or received from said SMBus ~~(213, 214)~~ interface while an instruction to ~~a "received~~ receive data ~~to"~~ at said block counter register ~~instruction~~ is executed, wherein ~~and~~ the ~~"received data to"~~ instruction to receive data at said block counter register from said SMBus interface ~~being~~ is completed when the value of said loop counter ~~(204)~~ reaches zero.

17. (Currently Amended) The integrated circuit chip of claim 10, wherein each instruction comprises one bit ~~(304)~~ indicating as to whether or not an instruction is the last instruction in the program.

18. (Currently Amended) The integrated circuit chip of claim 10, wherein each instruction comprises one bit ~~(305)~~ indicating as to whether an instruction is to be executed only once or this instruction is to be executed repeatedly until a loop counter ~~(204)~~ becomes zero, wherein said loop counter is decremented each time an instruction is executed repeatedly.

19. (Currently Amended) A ~~Method~~ method for controlling an SMBus, the method comprising:

identifying ~~(402)~~ a starting address of a program ~~(210, 211,212)~~ comprising one or more instructions; said program ~~(210, 211, 212)~~ being stored in a memory ~~(202)~~;

fetching instructions of said program one after another into a finite-state machine ~~(201)~~; and

transferring data between an SMBus ~~(213, 214)~~ interface and a register set in compliance with the instruction present in said finite-state machine (201).

20. (Currently Amended) The method ~~Method~~ of claim 19, wherein said register set complies with the ACPI specification.

21. (Currently Amended) The method ~~Method~~ of claim 19, wherein said identifying step comprises the sub-steps of:

reading a first value of a protocol register ~~(SMB_PRTCL)~~ specifying an offset in an address register array ~~(207)~~; and

reading a second value of a register of said address register array (207), said register being specified by said offset; said second value constituting said starting address of said program.

22. (Currently Amended) The method ~~Method~~ of claim 20, wherein said transferring step comprising the sub-steps of:

interpreting a ~~"received data to"~~ reception instruction;

reading the value of a buffer pointer register ~~(206)~~; and

transferring the data read from said SMBus ~~(213, 214)~~ interface to the data register ~~(SMB_DATA)~~ at which the value stored in said buffer pointer register ~~(206)~~ points.

23. (Currently Amended) The method ~~Method~~ of claim 22, wherein said transferring step further comprises incrementing said value of said buffer pointer register ~~(206)~~.

24. (Currently Amended) The method ~~Method~~ of claim 22, wherein said transferring step further comprising decrementing a loop counter ~~(204)~~ and checking as to whether said loop counter ~~(204)~~ has a value of zero.

25. (Currently Amended) The method ~~Method~~ of claim 20, wherein said transferring step comprises the sub-steps of:

interpreting a ~~"transmit data from"~~ transmission instruction;

reading the value of a buffer pointer register ~~(206)~~; and

transferring the data read from the data register ~~(SMB_DATA)~~ at which the value stored in said buffer pointer register ~~(206)~~ to said SMBus ~~(213, 214)~~ interface.

26. (Currently Amended) The method ~~Method~~ of claim 25, wherein said transferring step further comprises incrementing of said buffer pointer register ~~(206)~~.

27. (Currently Amended) The method of claim 25, wherein the transferring step further comprises decrementing of said loop counter ~~(204)~~.

28. (Currently Amended) The method of claim 19, wherein said transferring step comprises:

Interpreting an instruction to ~~a "~~transmit data from a block counter register ~~SMB_BCNT" instruction~~ to said SMBus interface;

storing the value of ~~a~~ said block count register ~~SMB_BCNT~~ in a loop counter ~~(204)~~; and

transmitting the value of said block ~~count~~ counter register ~~(SMB_BCNT)~~ to said SMBus ~~(213, 214)~~ interface.

29. (Currently Amended) The method of claim 19, wherein said transferring step comprises:

interpreting an instruction to ~~a "received~~ receive data ~~to SMB_BCNT" instruction~~ from said SMBus interface;

transmitting a byte from said SMBus ~~(213, 214)~~ interface to ~~a~~ said block count register ~~SMB_BCNT~~; and

storing the value of the byte received from said SMBus ~~(213, 214)~~ interface to a loop counter register ~~(204)~~.

30. (Currently Amended) The method of claim 19, wherein ~~the~~ said transferring ~~step~~ further comprises:

determining as to whether a stop bit ~~(304)~~ has a predetermined value; if this is the case:

writing 80h into a status register ~~(SMB_STS)~~ of said register set ~~(208)~~.

31. (Currently Amended) The method of claim 19, wherein ~~the~~ said transferring ~~step~~ further comprises:

determining as to whether a loop bit ~~(305)~~ of an instruction has a predetermined value; if that is the case,

executing said instruction repeatedly;

decrementing a loop counter ~~(204)~~ each time said instruction is executed;

finishing the execution of said loop instruction when the value of the said loop counter ~~(204)~~ becomes zero; and

fetching the next instruction.

32-95. (Cancelled).